viewment

Users

GET Get a profile page of user

```
/users/:userid/page
```

프로필 페이지

Request

Path Parameters

userid

OPTIONAL

`string`

Response

● 200: OK

```
{
    {
            "userId": "string",
            "nickname": "string",
            "email": "string",
            "message": "string",
            "profileImage": {
                                  "imageId": "string",
                                  "path": "string"
                            }
    }
    "countFollowings": 0,
    "countFollowers": 0,
    "countArticles": 0,
    "countMemories": 0,
    "articleImages": [
                       {
                          "imageId": "string",
                          "path": "string"
                        }
    ],
    followed: false
}
```

GET Search a user by email

/users/:email

Request

Path Parameters

email

OPTIONAL

`string`

Response

● 200: OK

```
{
        userId: "String",
        nickname: "String",
        followed: false,
        profileImage:{
             imageId: "String"
             path: "String"
        }
}
```

GET Search users by nickname

/users/like/:nickname

검색

Request

Path Parameters

nickname

OPTIONAL

`string`

Response

● 200: OK

```
{
    "content":[
        {
            "userId": "3zswbrmeRiUhPwi9FcYiQsLGlqn1",
            "nickname": "String",
            "profileImage": {
                imageId:"String",
                path:"String"
            },
            "followed": false
        }
    ],
    "pageable": "INSTANCE",
    "last": true,
    "totalPages": 0,
    "totalElements": 0,
    "size": 0,
    "number": 0,
    "sort":{
        "unsorted": true,
        "sorted": false,
        "empty": true
        },
    "numberOfElements": 0,
    "first": true,
    "empty": false
}
```

Account

POST sign in a new user

`/accounts`

Request

Path Parameters

userId
OPTIONAL
`string`

유저 아이디

Response

● 201: Created

GET Get an account of the user

`/accounts/:userId`

Request

Path Parameters

userId

OPTIONAL

`string`

유저아이디ㅇ

Response

● 200: OK

```
{
    userId: "String",
    nickname: "String",
    email: "String",
    message: "String",
    joinDate: "TimeStamp"
    profileImage:{
        imageId: "String",
        path: "String"
    }

}
```

GET 이메일 중복 확인

/accounts/:email/check

Request

Path Parameters

email

OPTIONAL

`string`

email

Response

● 200: OK

● 400: Bad Request

중복된 이메일

PATCH Modify an account of the user

/accounts/:userId

Request

Path Parameters

userId

OPTIONAL

`string`

유저아이디

Body Parameters

message

OPTIONAL

`string`

한줄요약

nickname

OPTIONAL

`string`

닉네임

Response

● 200: OK

DELETE Delete an user

`/accounts/:userId`

Request

Path Parameters

userId

OPTIONAL

`string`

Response

● 200: OK

```
```

Follows

POST Follow

```
/users/:userId/follow
```

Request

Path Parameters

userId

OPTIONAL

`string`

내 uid

Body Parameters

targetUserId

OPTIONAL

`string`

팔로우 할 uid

Response

● 200: OK

GET Get followers list

`/users/:userId/followers?page=0&size=10`

userId의 팔로워를 반환

Request

Path Parameters

userId

OPTIONAL

`string`

Query Parameters

size

OPTIONAL

`integer`

한번에 받을 팔로워의 수

page

OPTIONAL

`integer`

요청 횟수

Response

● 200: OK

```
{
    "content":[
        {
            "userId": "String",
            "nickname": "String",
            "profileImage": {
                imageId:"String",
                path:"String"
            },
            "followed": false
        }
    ],
    "pageable": "INSTANCE",
    "last": true,
    "totalPages": 0,
    "totalElements": 0,
    "size": 0,
    "number": 0,
    "sort":{
        "unsorted": true,
        "sorted": false,
        "empty": true
        },
    "numberOfElements": 0,
    "first": true,
    "empty": false
}
```

GET Get followings list

/users/:userId/followings?page=0&size=10

Request

Path Parameters

userId

OPTIONAL

`string`

Query Parameters

page

OPTIONAL

`integer`

상

size

OPTIONAL

`integer`

상

Response

● 200: OK

```
{
    "content":[
        {
            "userId": "String",
            "nickname": "String",
            "profileImage": {
                imageId:"String",
                path:"String"
            },
            "followed": false
        }
    ],
    "pageable": "INSTANCE",
    "last": true,
    "totalPages": 0,
    "totalElements": 0,
    "size": 0,
    "number": 0,
    "sort":{
        "unsorted": true,
        "sorted": false,
        "empty": true
        },
    "numberOfElements": 0,
    "first": true,
    "empty": false
}
```

DELETE Unfollow

/users/:userId/followings/:targetUserId

Request

Path Parameters

userId

OPTIONAL

`string`

내 uid

targetUserId

OPTIONAL

`string`

언팔로우할 uid

Response

● 200: OK

```
```

DELETE Remove a follower of the user by user id

`/users/:userId/followers/:targetUserId`

Request

Path Parameters

targetUserId

OPTIONAL

`string`

userId

OPTIONAL

`string`

Response

● 200: OK

Memories

POST Create Memory no pin

```
/users/:userId/memories
```

핀 없는 곳에 기억하기 만들 때

Request

Path Parameters

userId

OPTIONAL

`string`

Body Parameters

lat

OPTIONAL

`string`

위도

lng

OPTIONAL

`string`

경도

radius

OPTIONAL

`string`

반경 (m 단위로)

name

OPTIONAL

`string`

기억하기 이름

Response

● 200: OK

```
```

POST Create Memory with pin

/users/:userId/memories/pin

핀에다가 기억하기 만들 때

Request

Path Parameters

userid

OPTIONAL

`string`

Body Parameters

pinId

OPTIONAL

`string`

place 번호

radius

OPTIONAL

`string`

반경

name

OPTIONAL

`string`

기억하기 이름

Response

● 200: OK

GET

Get Memory list by userId

/users/:userId/memories

프로필에서 기억하는 장소 리스트 만들 때

Request

Path Parameters

userId

OPTIONAL

`string`

Response

● 200: OK

```
[
                {
                   "radius": 0,
                   "name": "string",
                   "pin": {
                             "pindId": 0,
                             "address": "string",
                             "lat": 0.0,
                             "lng": 0.0
                   }
                }
]
```

PUT Update Memory

/memories/:memoryId

기억하기 수정

Request

Path Parameters

memoryId

OPTIONAL

`string`

Body Parameters

memoryId

OPTIONAL

`string`

userId

OPTIONAL

`string`

pinId

OPTIONAL

`string`

radius

OPTIONAL

`string`

name

OPTIONAL

`string`

Response

● 200: OK

DELETE Delete Memory

/memories/:memoryId

Request

Path Parameters

memoryId

OPTIONAL

`string`

Response

● 200: OK

Pins

GET Get Pins

https://api.cakes.com/**pins**

This endpoint allows you to get free cakes.

Request

Response

● 200: OK

Cake successfully retrieved.

```
[
    {
        "pinId": "String",
        "address":"String",
        "lng": "String",
        "lat": "String"
    },
]
```

GET Get a pin

/pins/:pinId

Request

Path Parameters

pinId

OPTIONAL

`string`

pin Id

Response

● 200: OK

Articles

POST Write Article no pin

```
/articles
```

핀이 없는 곳에 게시글 작성, response data에 articleId가 반됨 .

Request

Body Parameters

userId

OPTIONAL

`string`

lat

OPTIONAL

`number`

lng

OPTIONAL

`number`

contents

OPTIONAL

`string`

hashtags

OPTIONAL

`array`

string 배

Response

● 200: OK

```
{
    0
}
```

POST Write Article with pin

/articles/pin

핀 있는 곳에 게시글 작성, response data에 articleId가 반환됨 .

Request

Body Parameters

userId

OPTIONAL

`string`

pinId

OPTIONAL

`number`

contents

OPTIONAL

`string`

pin id

hastags

OPTIONAL

`array`

string 배열

Response

● 200: OK

```
{
    0
}
```

GET Get Article by article id

/articles/:articleId

게시글 상세 보기 / 좋아요와 스크랩수 3차에서 구현

Request

Path Parameters

articleId

OPTIONAL

`number`

Response

● 200: OK

```
{
    "article": {
                  "articleId": 0,
                  "contents": "string",
                  "wdate": "string".
                  "user": {
                             "userId": "string",
                             "nickname": "string",
                             "profileImage": {
                                                "imageId": 0,
                                                "path" "string"
                             }
                  },
                  "pin": {
                            "pinId": "string",
                            "address": "string",
                            "lat": 0.0,
                            "lng": 0.0
                  },
                  "images": [
                               {
                                  "imageId": 0,
                                  "path": "string"
                               }
                  },
                  "hashtags": [
                                {
                                   "hashtagId": 0,
                                   "contents": "string"
                                }
                  ]
     }
     "replies": [
                   {
                      "replyId": 0,
                      "articleId": 0,
                      "contents": "string",
                      "wdate": "string",
                      "user": {
                                 "uerId": "string",
                                 "nickname": "string",
                                 "profileImage": {
                                                    "imageId": 0,
                                                    "path" "string"
                                 }
                      },
                      "rereplies": [
                                      {
                                         "replyId": 0,
                                         "articleId": 0,
                                         "contents": "string",
                                         "wdate": "string",
                                         "user": {
                                                    "userId": "strin
                                                     "nickname": "st
                                                     "profileImage":


                                                     }
                                         }
                                       }
                        ]
                     }
    ]
}
```

GET Get Articles With Thumbnail By User Id

/articles/searchbyuserid/:userId?page=0&size=0

프로필 페이지 그리드 화면 구성을 위해 게시글 id와 image의 path를 받 asd

Request

Path Parameters

OPTIONAL

`string`

Response

● 200: OK

```
```

GET Get Articles With Thumbnail By Pin Id

/articles/searchbypinid/:pinId?page=0&size=0

Request

Path Parameters

OPTIONAL

`string`

Response

● 200: OK

GET Get Articles With Thumbnail by Hashtag

/articles/searchbyhashtag/:hashtag?page=0&size=0

해시태그 검색 결과 (피드) - 해시태그를 검색에서 그리드 화면을 구성하기 위해 게시글 id와 썸네일 path를 받음 asd

Request

Path Parameters

hashtag

OPTIONAL

`string`

Response

● 200: OK

```
{
   [
      "articleId": 0,
      "image": {
                  "imageId": 0,
                  "path": 0
                }
   ]
}
```

PUT Update Article

/articles/:articleId

Request

Path Parameters

articleId

OPTIONAL

`string`

Body Parameters

userId

OPTIONAL

`string`

articleId

OPTIONAL

`number`

pinId

OPTIONAL

`number`

contnents

OPTIONAL

`string`

hashtags

OPTIONAL

`array`

string 배열

Response

● 200: OK

DELETE Delete Article

/articles/:articleId

Request

Path Parameters

articleId

OPTIONAL

`string`

Response

● 200: OK

Replies

POST Create Reply

```
/replies
```

Request

Body Parameters

userId

OPTIONAL

`string`

articleId

OPTIONAL

`string`

rereplyId

OPTIONAL

`string`

대댓글일 경우 댓글의 rid 대댓글이 아닐 경우 0

contents

OPTIONAL

`string`

Response

● 200: OK

PUT Update Reply

`/replies/:replyId`

Request

Path Parameters

replyId

OPTIONAL

`string`

Body Parameters

userId

OPTIONAL

`string`

contents

OPTIONAL

`string`

Response

● 200: OK

DELETE Delete Reply

`/replies/:replyId`

Request

Path Parameters

replyId

OPTIONAL

`string`

Response

● 200: OK

Hashtags

GET Get Hashtags by Keyword

`/hashtags/:keyword`

Request

Path Parameters

keyword

OPTIONAL

`string`

Response

● 200: OK

```
{
    "hashtags": [
                    {
                       "hid": 0,
                       "contents": "string"
                    }
                 ]
}
```

Images

POST Upload Article Images

```
/images/article/:articleId
```

게시글에 포함되어 있는 이미지 저장

Request

Path Parameters

articleId

OPTIONAL

`string`

Headers

Content-type

OPTIONAL

`string`

multipart/form-data

X-Authorization-Firebase

OPTIONAL

`string`

jwt token

Form Data Parameters

articleImages

OPTIONAL

`object`

Response

● 200: OK

POST Upload Profile Image

`/images/profile/:userId`

Request

Path Parameters

userId

OPTIONAL

`string`

Headers

Content-type

OPTIONAL

`string`

multipart/form-data

X-Authorization-Firebase

OPTIONAL

`string`

jwt-token

Form Data Parameters

profileImage

OPTIONAL

`string`

Response

● 200: OK

GET Get Image

/images/:path

path 예시) 프로필사진 "profile/{uid}" 게시글 사진 "article/{articleId}-{no}" (no는 사진 순서)

.

Request

Path Parameters

path

OPTIONAL

`string`

Response

● 200: OK

```
```

PUT Update Profile Image

`/images/profile/:userId`

Request

Path Parameters

userId

OPTIONAL

`string`

Headers

Content-type

OPTIONAL

`string`

X-Authorization-Firebase

OPTIONAL

`string`

Form Data Parameters

profileImage

OPTIONAL

`string`

Response

● 200: OK

```
```

DELETE Delete Profile Image

/images/profile/:userId

Request

Path Parameters

userId
OPTIONAL
`string`

Headers

X-Authorization-Firebase
OPTIONAL
`string`

Response

● 200: OK

```
```